Kernel-level Thread Scheduling

Introduction:

The source code has 2 ways of accommodating threads in CPU – one with scheduler and one without. In the second case each thread handles the CPU to other thread on its own. The following document shows how the scheduler is implemented.   
Two kinds of scheduler are implemented: 1) FIFO and 2) Round-Robin scheduler.

Implementation Overview and code elaboration:

Scheduler class has member functions like enqueue, dequeue, add, resume, yield and terminate. It has a member variable which are head, rear of the ready queue (declared as static, since all the functions are supposed to use them) and finally struct which is basically the node of the ready-queue. It has the thread id and pointer to the next node.

Modifications have been made to thread\_shutdown of the thread terminate function to handle the termination of the thread.

1. Enqueue: It adds node to the end of the ready\_queue.
2. Dequeue: this function removes the head of the queue and returns it
3. Add: it calls the enqueue function to add the thread into the queue.
4. Resume: it calls the enqueue function to add the thread into the queue.
5. Yield: this function calls the dequeue , gets the head of the queue and sends that as the argument to dispatch\_to function of thread class.
6. Terminate: This function calls the yield and deletes the thread that is called to be terminated.

Now the thread function – thread\_shutdown is modified so that it calls the terminate function of the thread.

Bonus option1:

1. Enabling and disabling the interrupts at the right places.   
   Since we are pushing 0 for the IF flag in setup\_context which is called in the thread class constructor, the interrupts are disabled. So once it is out of the constructor, we are enabling the interrupts in the thread\_start function.

Since we don’t want atomic violations, we are disabling at the start of the resume and enabling them again at the end of yield. Also we are only disabling wen the interrupts are enabled.

1. Round-Robin Scheduler:  
   The only changes made to the fifo are   
   1. Interrupts included in the resume and yield functions

2. Modified the interrupt handler in the simple\_timer class. For every 50ms, the scheduler yields the current thread and adds it behind the queue.